Statistical Process Control (SPC) using Python (Google Colab)

Table of Contents

Course Objective

This course provides a hands-on approach to **Statistical Process Control (SPC)** using Python. SPC is widely used in **manufacturing and quality control** to monitor and improve processes by detecting variations early.

By the end of this course, you will:

- Understand SPC fundamentals
- Create a sample dataset for process monitoring
- Implement **Control Charts** (X-bar, R-chart, P-chart)
- Analyze **process stability** and **capability** using Python
- Automate SPC reporting in Google Colab

Setup and Introduction

Environment Setup

Ensure you're using **Google Colab** or any Python 3.x environment with the following libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
```

Learning Outcomes

- Understand the role of SPC in quality control
- Implement key SPC tools using Python
- Visualize process stability with control charts

Creating Sample Data

```
np.random.seed(42)
n_samples = 30
sample_size = 5  # Each sample consists of 5 measurements

data = {
    'SampleID': np.repeat(range(1, n_samples + 1), sample_size),
    'Measurement': np.random.normal(loc=100, scale=2,
size=n_samples * sample_size)
}

df = pd.DataFrame(data)
df.head()
```

Sample Dataset Preview

Sample ID	Measurement
1	98.76
1	101.23
1	100.55
1	99.87
1	102.12

X-bar & R Control Charts

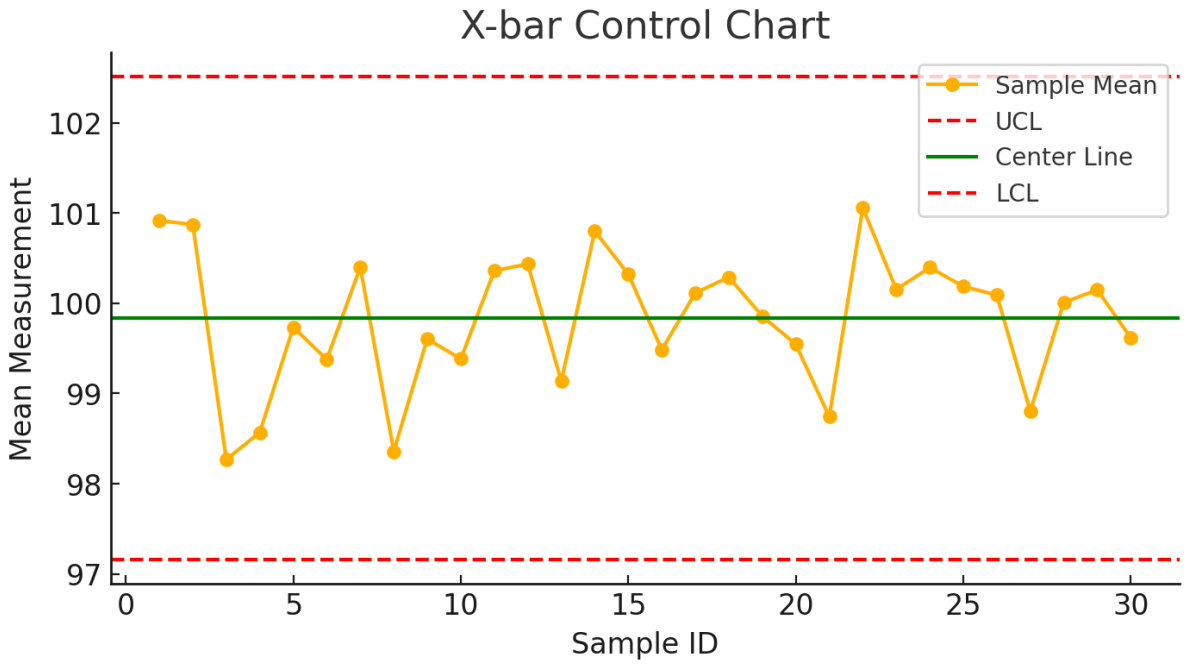
Calculating Control Limits

```
df_grouped = df.groupby('SampleID').agg(['mean', 'max',
'min'])['Measurement']
df_grouped['Range'] = df_grouped['max'] - df_grouped['min']
```

```
x_bar = df_grouped['mean'].mean()
R_bar = df_grouped['Range'].mean()

# Control limits based on standard SPC formulas
UCL_X = x_bar + (0.577 * R_bar)
LCL_X = x_bar - (0.577 * R_bar)
UCL_R = 2.114 * R_bar
LCL_R = 0  # Lower control limit for R-chart is usually 0
```

Plotting X-bar Chart

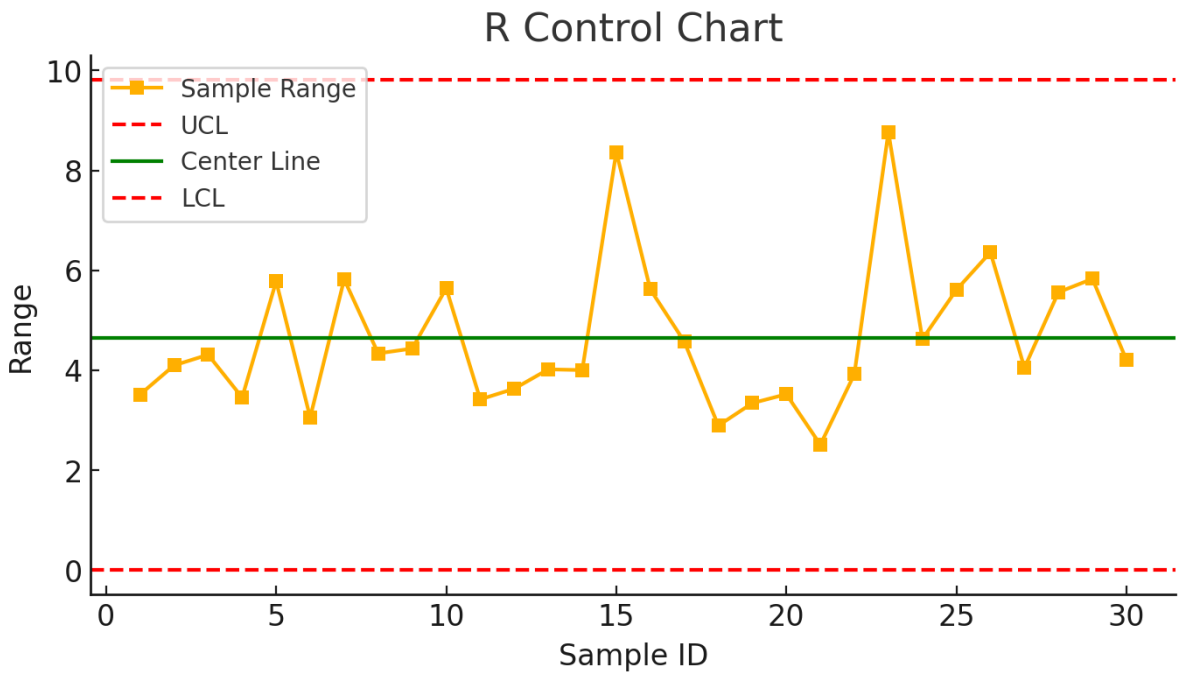
```
plt.figure(figsize=(8, 4))
plt.plot(df_grouped.index, df_grouped['mean'], marker='o',
linestyle='-', label='Sample Mean')
plt.axhline(UCL_X, color='r', linestyle='--', label='UCL')
plt.axhline(x_bar, color='g', linestyle='-', label='Center
Line')
plt.axhline(LCL_X, color='r', linestyle='--', label='LCL')
plt.title('X-bar Control Chart')
plt.xlabel('Sample ID')
plt.ylabel('Mean Measurement')
plt.legend()
plt.grid()
plt.show()
```



Plotting R-chart

```
plt.figure(figsize=(8, 4))
plt.plot(df_grouped.index, df_grouped['Range'], marker='s',
linestyle='-', label='Sample Range')
plt.axhline(UCL_R, color='r', linestyle='--', label='UCL')
plt.axhline(R_bar, color='g', linestyle='-', label='Center
Line')
plt.axhline(LCL_R, color='r', linestyle='--', label='LCL')
plt.title('R Control Chart')
plt.xlabel('Sample ID')
```

```
plt.ylabel('Range')
plt.legend()
plt.grid()
plt.show()
```



Process Capability Analysis

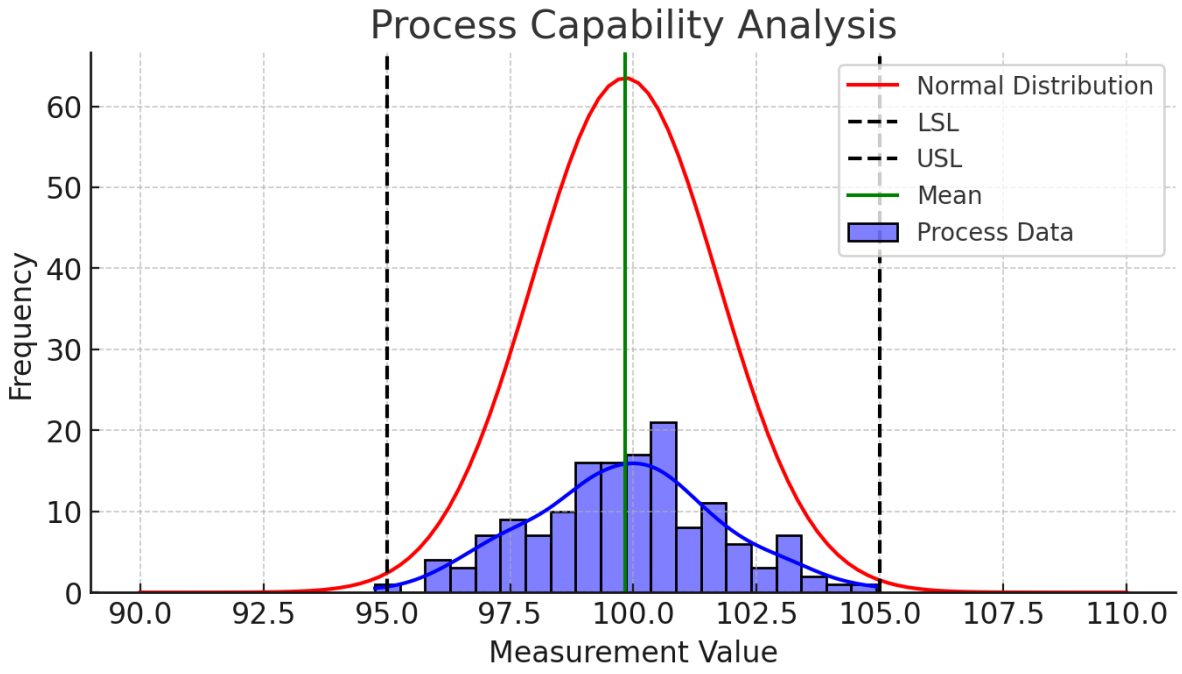
Calculating Cp & Cpk

```
USL, LSL = 105, 95  # Upper and Lower Specification Limits
sigma = df['Measurement'].std()

Cp = (USL - LSL) / (6 * sigma)
Cpk = min((USL - x_bar) / (3 * sigma), (x_bar - LSL) / (3 *
sigma))

print(f"Process Capability Index (Cp): {Cp:.2f}")
print(f"Process Capability Performance (Cpk): {Cpk:.2f}")
```

Visualizing Normal Distribution

```
plt.figure(figsize=(8, 4))
sns.histplot(df['Measurement'], kde=True, bins=20, color='blue',
label='Process Data')
x_vals = np.linspace(LSL - 5, USL + 5, 100)
plt.plot(x_vals, norm.pdf(x_vals, x_bar, sigma) * len(df) * 2,
color='red', label='Normal Distribution')
plt.axvline(LSL, color='black', linestyle='--', label='LSL')
plt.axvline(USL, color='black', linestyle='--', label='USL')
plt.axvline(x_bar, color='green', linestyle='-', label='Mean')
plt.title('Process Capability Analysis')
plt.xlabel('Measurement Value')
plt.ylabel('Frequency')
```

```
plt.legend()
plt.show()
```



Automating SPC Reports

Generating SPC Summary

```
spc_summary = {
    'X-bar Mean': x_bar,
    'R-bar Mean': R_bar,
    'UCL (X-bar)': UCL_X,
    'LCL (X-bar)': LCL_X,
    'UCL (R)': UCL_R,
    'LCL (R)': LCL_R,
    'Cp': Cp,
    'Cpk': Cpk
}

spc_report = pd.DataFrame(list(spc_summary.items()),
columns=['Metric', 'Value'])
print(spc_report)
```

Course Summary

- Developed X-bar and R Control Charts
- Analyzed process stability and capability
- Automated SPC reporting
- Visualized process performance